1 Getting Up and Running

To begin with, make sure that you are using the IPython shell with the PyLab extension. At the time of writing, in a Unix environment, you would type

```
ipython -pylab
```

This will set up an interactive environment that behaves similarly to the MATLAB environment.

Now, we will import some packages that we will need to perform basic operations. You should consult the Python tutorial to get a better understanding of what is happening. Type

```
import scipy
```

2 Loading and Viewing Your First Image

Now we are ready to load an image. The PyLab package, which is automatically imported when you launch IPython with the "-pylab" flag, can only import PNG images. (Note: This may have changed with Matplotlib 0.98). So, we will use the `imread` function from the `scipy` package:

```
img = scipy.misc.imread("einstein.png")
```

As you get more familiar with Python, you will learn how to shorten the length of that line by importing the imread function into the main namespace. Again, the reader is referred to the Python tutorial.

Now, let's look at that image:

```
imshow(img)
```

The Matplotlib package treats the image as just an array of numbers and pseudo-colors it. If we want the image to be displayed in grayscale, we would type:

```
gray()
```

This sets the colormap of the image to be grayscale.

3 Basic Filtering

Now that we have the image, we'll probably want to begin with some basic filtering. Most of basic filtering functions are in the `scipy.signal` package, so we'll need to import it:

```
import scipy.signal
```

Let's convolve the image with a derivative filter:

```
filtered_img = scipy.signal.convolve2d(img,[[-1,1]],"valid")
imshow(filtered_img)
```

In particular, I would like to clarify what the command `[[-1,1]]` means. Python, unlike MATLAB, does not allow you to create arrays or matrices as literal values in your code. So, `[[-1,1]]` is a list containing the list `[-1,1]`. To see how NumPy treats this, you can type `array([[-1,1]])` to see how this is converted into an array. To see more, try the command `array([[-1,1]]).shape` and `array([-1,1]).shape` and compare the difference.

4 Writing Your First Script

Of course, you will not want to type in commands every time. IPython has a nice feature that you can use to run scripts. Typing `run`¡*script name*¿ will run the script, then add the variables to the root namespace. To see this, type the following script in:

```
import scipy,scipy.signal
from pylab import *

img = scipy.misc.imread("einstein.png")
filtered_img = scipy.signal.convolve2d(img,[[-1,1]],"valid")
imshow(filtered_img)
gray()
show()
```

In all the scripts, you will need to import the numerical libraries as the run commands starts with a fairly empty namespace.

5 From Here

At this point, you should consult information available at `http://www.scipy.org` .You should also read the Python tutorial at `http://www.python.org` .